

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería Informática**

## **TRABAJO FIN DE GRADO**

**ANOTACIÓN Y CLASIFICACIÓN DE EVENTOS PARA LA  
MONITORIZACIÓN CONTINUA DE RUTINAS HUMANAS  
MEDIANTE SENSORES NO INVASIVOS**

**Sergio Arranz Moya**  
**Tutor: Pablo Varona Martínez**

**ENERO 2021**



# **ANOTACIÓN Y CLASIFICACIÓN DE EVENTOS PARA LA MONITORIZACIÓN CONTINUA DE RUTINAS HUMANAS MEDIANTE SENSORES NO INVASIVOS**

**AUTOR: Sergio Arranz Moya**  
**TUTOR: Pablo Varona Martínez**

**Dpto. de Ingeniera Informática**  
**Escuela Politécnica Superior**  
**Universidad Autónoma de Madrid**  
**Enero de 2021**





# Resumen

Las narices artificiales son dispositivos electrónicos no invasivos cuyo objetivo es detectar distintos gases en el ambiente en el que se encuentran, simulando a una nariz humana. Las señales que se registran en los sensores de dichas narices no son invariantes en el tiempo, por lo que su valor depende no solo de condiciones actuales sino también de las pasadas. Esto otorga temporalidad a los datos que se obtienen de estos dispositivos.

En este Trabajo de Fin de Grado exploramos distintos problemas de clasificación de series temporales de narices artificiales. Cada uno de ellos corresponde a una rutina humana distinta, concretamente relacionadas con el ámbito escolar, e intentamos detectar rutinas en base a la estructura temporal de los gases captados por los distintos sensores.

Debido a que tratamos con temporalidad en los datos, nos encontramos frente a problemas no triviales como variabilidad y el ruido de las señales. También hay que tener en cuenta que utilizamos medidas simultáneas de distintos gases, por lo que las series temporales para la clasificación son multivariadas, aunque analizaremos también series monovariadas, utilizando cada gas individualmente para caracterizar la información que contienen.

Basándonos en trabajos de investigación de los últimos años dedicados a la exploración de métodos de clasificación para series temporales, seleccionaremos y adaptaremos los más prometedores para nuestro objetivo. También investigaremos los posibles métodos de codificación y etiquetado en dichas series.

Tras la realización de las diversas pruebas en los distintos problemas, expondremos las conclusiones sobre cada uno de ellos y, de forma más general, sobre la clasificación de rutinas humanas mediante sensores no invasivos.

# Abstract

Artificial noses are non-invasive electronic devices whose objective is to detect different gases in the environment where they are located, simulating a human nose. The signals registered by the sensors of said noses are not time-invariant, so its value depends not only on the actual conditions but also on the past ones. This grants temporality to the data obtained from these devices.

In this bachelor thesis we explore different classification problems of time series from artificial noses. Each of them corresponds to a unique human routine, specifically related to the school environment, so we will be trying to detect routines based on the time structure of the gases captured by the different sensors.

Taking into account we are dealing with temporality in the data, we are facing non-trivial problems like variability and the noise of the signals. We also have to keep in mind that we use measurements of several gases, so the time series that we have for classification are multivariate, although we will also analyze univariate series, using each gas individually to characterize the information they contain.

Based on research work in recent years dedicated to exploring classification methods for time series, we will select the most promising ones for our objective. We will also research on the possible codification and labeling for said series.

After carrying out the various tests on the different problems, we will present the conclusions for each of them, and more generally, about classification of human routines with non-invasive sensors.

## **Palabras clave**

Narices artificiales, Clasificación de series temporales, Series temporales multivariadas, Etiquetado de series temporales, Monitorización no invasiva de actividad humana.

## **Keywords**

Artificial noses, Time series classification, Time series tagging, Multivariate time series, Non-invasive human activity monitoring.





## ***Agradecimientos***

A mi familia, por apoyarme todo este tiempo.

A mis amigos, por estar siempre ahí.

A mi tutor, por guiarme durante todo este trabajo.



## INDICE DE CONTENIDOS

1 Introducción.....	1
1.1 Series Temporales.....	1
1.2 El problema de la clasificación de series temporales .....	2
1.3 Codificación y etiquetado .....	4
1.4 Series temporales procedentes de narices artificiales.....	6
1.5 Objetivos del TFG .....	7
1.6 Organización de la memoria.....	7
2 Diseño y desarrollo.....	8
2.1 Estrategias de codificación .....	8
2.1.1 Random Forest basado en SMTS .....	8
2.1.2 Algoritmos derivados de red convolucional.....	9
2.2 Estrategias de etiquetado .....	9
2.3 Algoritmos de clasificación empleados en este TFG .....	10
2.3.1 Random Forest basado en SMTS .....	10
2.3.2 Fully convolutional network (FCN) .....	11
2.3.3 Red profunda residual (RNN) .....	12
2.4 Descripción de las series temporales de la nariz artificial.....	13
2.5 Metodología de análisis.....	14
3 Resultados.....	17
3.1 Problema de días lectivos .....	18
3.2 Problema de periodos lectivos.....	19
3.3 Problema de días de la semana.....	21
3.4 Problema de asignaturas .....	22
3.5 Problema de franjas horarias .....	26
3.6 Problema de días peculiares .....	27
3.7 Problema de horas de la limpieza.....	28
4 Conclusiones y discusión .....	29
4.1 Conclusiones.....	29
4.2 Trabajo futuro .....	30
Referencias .....	33
Glosario .....	35
Anexo A: Calendario escolar.....	- 1 -

# INDICE DE FIGURAS

FIGURA 1-1: EJEMPLO DE SERIE TEMPORAL DE UNA SOLA VARIABLE. ....	1
FIGURA 1-2: ARQUITECTURA DE RED CONVOLUCIONAL.....	3
FIGURA 1-3: ARQUITECTURA DE RED RESIDUAL.....	3
FIGURA 1-4: EJEMPLO DE SHAPELETS. ....	5
FIGURA 2-1: EJEMPLO DE ÁRBOL DE DECISIÓN DEL RANDOM FOREST PARA LA CODIFICACIÓN DE SERIES TEMPORALES.....	8
FIGURA 2-2: ARQUITECTURA DE FCN UTILIZADA EN ESTE TRABAJO.....	11
FIGURA 2-3: ARQUITECTURA DE RED RESIDUAL UTILIZADA. PANEL SUPERIOR: ARQUITECTURA DE RED. PANEL INFERIOR: ARQUITECTURA DE BLOQUE. SE APLICAN 64 FILTROS PARA EL BLOQUE 1 Y 128 PARA LOS BLOQUES 2 Y 3.....	12
FIGURA 2-4: EJEMPLO DE ALINEACIÓN DE SEÑALES EN LOS SENSORES (TGS2600).....	14
FIGURA 3-1: NIVEL DE CO <sub>2</sub> DURANTE CADA DÍA DE CURSO, DIVIDIDOS EN LECTIVOS Y NO LECTIVOS .....	19
FIGURA 3-2: NIVEL DE HUMEDAD DURANTE CADA DÍA DE CURSO, DIVIDIDOS EN LECTIVOS Y NO LECTIVOS .....	19
FIGURA 3-3: GRÁFICAS DE CO <sub>2</sub> EN CLASES DE LENGUA Y MÚSICA.....	24
FIGURA 3-4: GRÁFICAS DE CO <sub>2</sub> EN CLASES DE LENGUA Y VALORES .....	25
FIGURA 3-5: GRÁFICAS DE CO <sub>2</sub> EN CLASES DE LENGUA Y MATEMÁTICAS .....	26

## INDICE DE TABLAS

TABLA 2-1: ATRIBUTOS DE LAS SERIES TEMPORALES .....	13
TABLA 3-1: PARÁMETROS RANDOM FOREST BASADO EN SMTS. ....	17
TABLA 3-2: PARÁMETROS FCN. ....	17
TABLA 3-3: PARÁMETROS RNN.....	18
TABLA 3-4: % DE ACIERTO PROBLEMA DE DÍAS LECTIVOS .....	18
TABLA 3-5: % DE ACIERTO PROBLEMA DE PERIODOS LECTIVOS.....	20
TABLA 3-6: % DE ACIERTO PROBLEMA DE PERIODOS LECTIVOS REDUCIDO .....	20
TABLA 3-7: % DE ACIERTO PROBLEMA DE DÍAS DE LA SEMANA.....	21
TABLA 3-8: PRECISIÓN Y SENSIBILIDAD (EN %) DE CADA DÍA DE LA SEMANA PARA SMTS UTILIZANDO ÚNICAMENTE LOS NIVELES DE CO <sub>2</sub> .....	22
TABLA 3-9: % DE ACIERTO PROBLEMA DE ASIGNATURAS .....	22
TABLA 3-10: PRECISIÓN Y SENSIBILIDAD (EN %) DE CADA ASIGNATURA PARA SMTS UTILIZANDO ÚNICAMENTE LOS NIVELES DE CO <sub>2</sub> .....	23
TABLA 3-11: % DE ACIERTO PROBLEMA DE ASIGNATURAS (SOLO LENGUA Y MÚSICA) .....	23
TABLA 3-12: % DE ACIERTO PROBLEMA DE ASIGNATURAS (SOLO LENGUA Y VALORES).....	24
TABLA 3-13: % DE ACIERTO PROBLEMA DE ASIGNATURAS (SOLO LENGUA Y MATEMÁTICAS).....	25
TABLA 3-14: % DE ACIERTO PROBLEMA DE FRANJAS HORARIAS.....	26
TABLA 3-15: PRECISIÓN Y SENSIBILIDAD (EN %) DE CADA FRANJA HORARIA PARA SMTS UTILIZANDO ÚNICAMENTE LOS NIVELES DE CO <sub>2</sub> .....	27
TABLA 3-16: % DE ACIERTO PROBLEMA DE DÍAS PECULIARES .....	27
TABLA 3-17: % DE ACIERTO PROBLEMA DE HORAS DE LA LIMPIEZA .....	28

# 1 Introducción

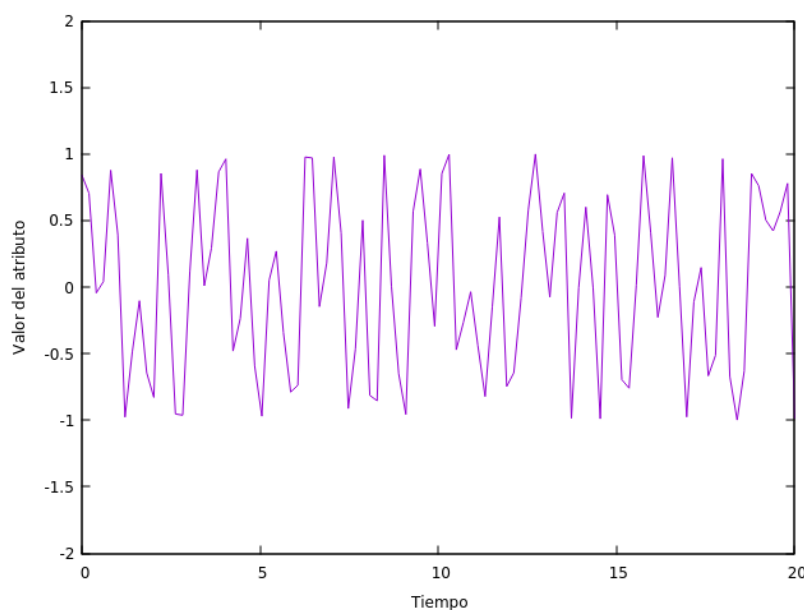
---

## 1.1 Series Temporales

Una serie temporal es una sucesión de datos de longitud finita cuya ordenación es relevante, ya sea porque indica el orden en el que han sido obtenidos (es decir, les dota de temporalidad y secuencialidad) o por cualquier otra razón pertinente. La Figura 1-1 muestra una serie temporal donde podemos ver el desarrollo de un atributo a través del tiempo. Si consideramos un único punto de ese grafico no obtenemos mucha información, pero si utilizamos un subconjunto de ellos y analizamos la estructura de la evolución temporal de la serie completa es posible observar la tendencia en ese momento y utilizar esa información para clasificar los datos o predecir futuros valores mediante distintos métodos. Dependiendo del número de atributos de los que están compuestas, estas series pueden ser o monovariadas o multivariadas.

Por un lado, las series temporales monovariadas consisten en la evolución de un único atributo a través del tiempo. Se puede ver como un dato compuesto de distintos valores formando un vector  $X = [x_1, x_2, x_3, \dots, x_t]$  donde cada  $x_t$  corresponde al valor del atributo en el tiempo  $t$ . La figura 1 forma parte de este grupo.

Por otro lado, las series multivariadas tienen mayor complejidad. Como su nombre indica, cada dato tomado en el tiempo consta de distintos atributos, por lo que tratar estas series igual que las monovariadas no es deseable ya que un grupo de atributos puede contener información relevante en su conjunto para el problema abordado y por tanto requiere ser manejado teniendo en cuenta esta diferencia respecto de un atributo individual. En las series multivariadas la información secuencial entre atributos puede ser de gran importancia para resolver el problema abordado. De forma general, una serie multivariada se representa típicamente como una matriz  $M = [X_1, X_2, X_3, \dots, X_i]$  donde cada  $X$  es una serie monovariada correspondiente a uno de los atributos.



**Figura 1-1: Ejemplo de serie temporal de una sola variable.**

## **1.2 El problema de la clasificación de series temporales**

Al igual que con conjuntos de datos estáticos, los problemas de series temporales que pueden encontrarse se centran fundamentalmente en dos tipos. Los problemas de regresión que buscan predecir futuros valores de las series, y los problemas de clasificación en los que nos centraremos, que intentan asignar a distintas series una clase de entre varias ya establecidas.

El interés en clasificar una serie temporal reside en que existe la posibilidad de que dado un problema de clasificación pueda encontrarse una correlación entre los atributos, el tiempo en el que fueron captados y la clase a la que pertenece cada serie. En estos casos realizar clasificación de datos estáticos resultaría mucho menos eficaz y en bastantes situaciones, totalmente inútil. Pese a su relevancia y la existencia de multitud de documentos acerca de este tipo de problemas de clasificación, hasta hace pocos años la gran complejidad de este problema y la metodología de pruebas utilizada daban lugar a resultados relativamente pobres (Rozado et al., 2010; Montgomery et al., 2015).

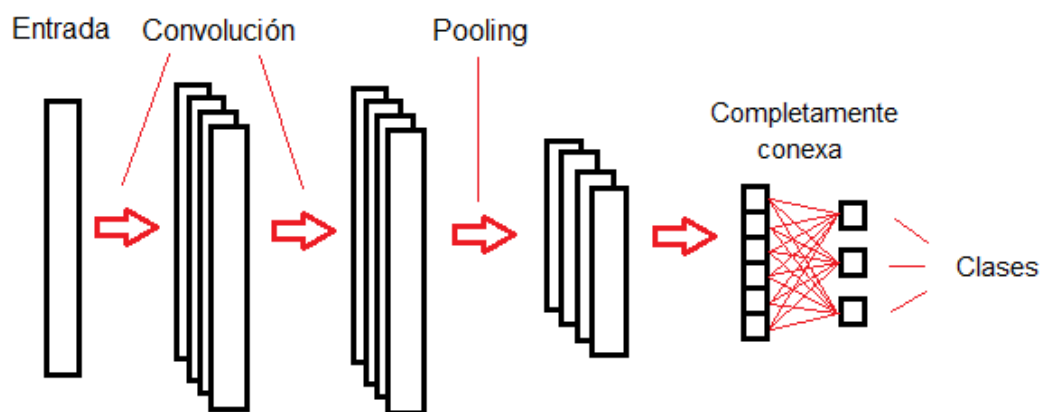
Las aplicaciones de la clasificación de series temporales han aumentado bastante en los últimos años debido al incremento de conjuntos de datos con información temporal creados en distintas áreas, fundamentalmente con la expansión del *big data*. En el campo de las finanzas, las series temporales se crean continuamente con respecto a cifras de ventas, situación de la bolsa, análisis de presupuestos etc., las cuales pueden dar información sobre la situación económica actual y futura o eventos remarcables (Andersen et al., 2009). En medicina, las series temporales pueden ayudar a poder detectar y prevenir distintas enfermedades y condiciones médicas (Zeger et al., 2006). Con respecto a multimedia, se genera una cantidad ingente de información relacionada con imágenes y sonidos (Rozado et al., 2012). La ciberseguridad, la física y electrónica son otros de los muchos campos que se benefician de la clasificación de series temporales, así como en otros ámbitos como las redes sociales, las redes de sensores, etc. (Small, 2005).

En comparación con la clasificación de datos estáticos, la clasificación de series temporales tiene la dificultad añadida de que hay que tener en cuenta la temporalidad de los datos. Esto requiere de algoritmos de clasificación que respeten el orden en el que se producen, ya que es el que dota a los datos de esta temporalidad y secuencialidad. Por esta razón se descartan diversos enfoques convencionales en los cuales no se tiene en cuenta de forma alguna la ordenación temporal de los datos.

Para llevar a cabo esta clasificación, se han explorado distintos tipos de algoritmos y variantes con resultados muy prometedores. Pese a la falta de temporalidad de las redes neuronales profundas comunes, se ha explorado el uso de redes convolucionales y residuales, otros algoritmos de aprendizaje profundo con arquitecturas más complejas que son capaces de respetar la temporalidad (Fawaz et al., 2019).

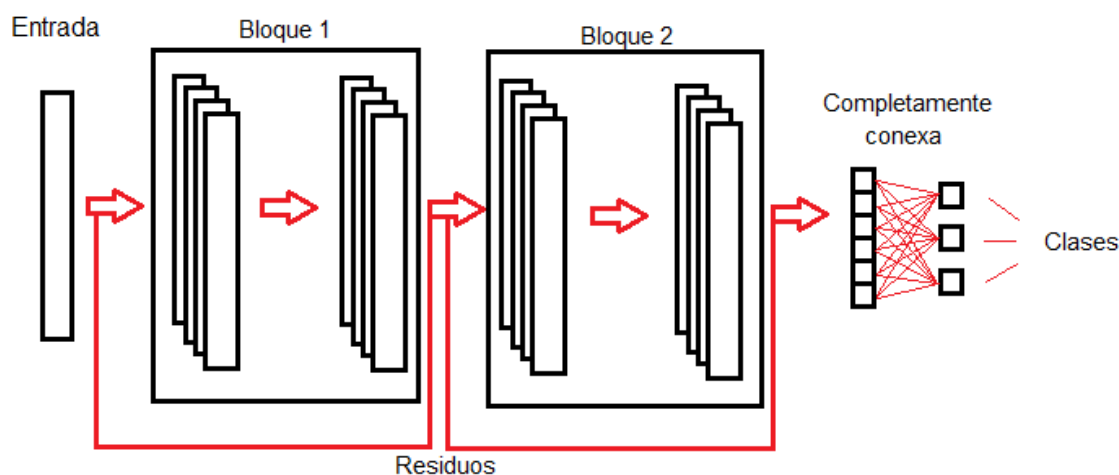
Las redes convolucionales se basan en la operación llamada convolución, que consiste en aplicar un filtro desplazable sobre los datos de entrada. Aunque por lo general estos filtros se crean para procesar matrices bidimensionales, para las series temporales monovariadas se utilizan filtros unidimensionales ya que el filtrado se realiza únicamente sobre el tiempo (Zhao et al., 2017).





**Figura 1-2: Arquitectura de red convolucional.**

Una red de este tipo suele estar compuesta de varias capas convolucionales, y puede contener capas de pooling, que se encargan de reducir el tamaño de los datos y extraer características dominantes. Estas redes suelen acabar con una capa clasificadora como puede ser una capa neuronal completamente conexa que es la encargada de predecir la clase del dato. En la Figura 1-2 se puede observar un ejemplo de esta arquitectura. Si la red no dispone de capas de pooling y por tanto el tamaño original de la entrada no se modifica, se considera una red completamente convolucional (FCN).



**Figura 1-3: Arquitectura de red residual.**

Las redes residuales por su parte también hacen uso de la convolución, con la diferencia de que cada bloque de capas convolucionales delimitado utiliza como entrada no solo la salida del bloque anterior sino también de la entrada de dicho bloque. Este residuo se utiliza para reducir el llamado problema de desvanecimiento del gradiente, que consiste en que a medida que se avanza en el proceso de aprendizaje, las actualizaciones de pesos van disminuyendo hasta llegar incluso a parar el entrenamiento. En la Figura 1-3 se puede observar una arquitectura de este tipo de red. Estas redes han mostrado rendimientos de nivel de estado del arte en otros campos como por ejemplo el reconocimiento de objetos. Estas arquitecturas complejas de aprendizaje profundo han obtenido resultados muy competitivos (Ismail Fawaz et al., 2019).

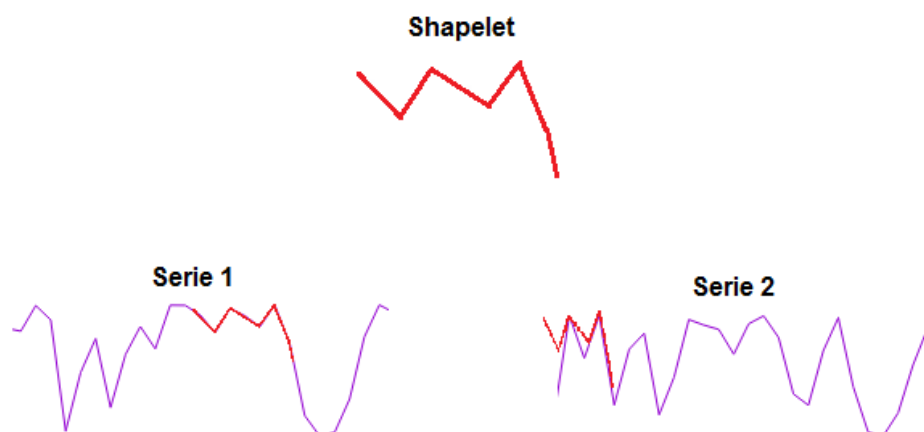
También se han explorado en profundidad los métodos basados en distancias. En estos métodos se definen distintos tipos de similitudes que se introducen de distintas formas a algoritmos de clasificación basados en distancia, siendo el más conocido k-NN. La exploración de distintas similitudes basadas en la naturaleza temporal de las series ha sido el principal enfoque de investigación, pero en los últimos años la búsqueda de algoritmos complejos que sean capaces de aprovechar dichas similitudes y obtener un rendimiento comparable o superior a kNN ha obtenido resultados muy competitivos (Abanda et al., 2019).

Entre otros tipos de algoritmos podemos encontrar aquellos basados en *shapelets* que son pequeños patrones que pueden aparecer en cualquier punto de las series y ayudan a definir cada clase mediante su ausencia o presencia, los basados en intervalos que utilizan partes de la serie en vez de al completo, los basados en modelos que crean para cada serie un modelo con el que se realizan comparaciones por similitud de forma similar a los métodos basados en distancias, los basados en diccionarios cuyo concepto es similar a los basados en *shapelets* pero utilizan frecuencia en vez de ausencia o presencia etc. También se han explorado combinaciones de estos tipos de algoritmos que han dado lugar a algoritmos con rendimientos vanguardistas (Bagnall et al., 2017).

Debido a la complejidad de las series temporales multivariadas, también se han explorado métodos especializados en ellas. Un ejemplo de esto es el uso de varios árboles de decisión para crear una representación simbólica de las series que intenta preservar su estructura temporal (SMTS) (Baydogan and Runger, 2014) que tiene en cuenta todos los atributos de las MTS al mismo tiempo. Esta representación simbólica se crea debido a la necesidad de extraer las relaciones entre los atributos, las cuales no se tienen en cuenta en métodos orientados a series temporales de forma más general ya que dicha relación no existe en las series temporales monovariadas. Tras generar la SMTS, se utiliza el enfoque conocido como bolsa de palabras, en el cual se tiene en cuenta la frecuencia relativa de cada símbolo para realizar la clasificación de la serie.

### **1.3 Codificación y etiquetado**

Para su uso en clasificación, las series temporales pueden ser modificadas de distintas formas aportándoles así un mejor uso y mayor eficacia a los clasificadores donde se introducen. El uso de la serie entera es un enfoque simple del que se hace uso en bastantes métodos basados en distancia ya que pueden existir características discriminatorias que dependen del punto de la serie en el que se encuentren. Otro enfoque es el uso de intervalos, con el cual se puede reducir el impacto que tiene el ruido en el clasificador y además resulta beneficioso cuando el problema tiene regiones discriminatorias.



**Figura 1-4: Ejemplo de shapelets.**

La búsqueda de *shapelets* es otro de los métodos de codificación comúnmente utilizados. Los *shapelets* son patrones inusuales que pueden encontrarse a lo largo de la serie cuya ausencia o presencia puede utilizarse para identificar la clase concreta (Ye and Keogh, 2011). En la Figura 1-4 se muestra un *shapelet* y dos series donde en la primera se encuentra un patrón muy similar y en la segunda no, dándonos información de a qué clase pertenece cada una. De forma similar, Los diccionarios también se aprovechan de estos patrones, pero en vez de fijarse en si existen en la serie, capturan la frecuencia con la que aparecen.

El etiquetado de una serie consiste en la asignación a cada dato de una serie de etiquetas descriptivas en base a su contenido. Estas etiquetas contienen alta cantidad de información sobre el dato en cuestión, y son útiles para diversas tareas como facilitar la clasificación, contextualizarla o usarla en sistemas de recomendación.

Este etiquetado puede llevarse a cabo manual o automáticamente. Como su nombre indica el etiquetado manual consiste en asignarle etiquetas a cada dato uno por uno sin uso de ningún algoritmo. El etiquetado automático es una tarea de clasificación que intenta estimar las etiquetas correspondientes de cada dato en base a sus atributos. Depende de la complejidad del etiquetado, este se puede llevar a cabo desde algoritmos simples a redes neuronales. Este tipo de automatización se basa en la conocida como extracción de características o *features*, que consiste en la extracción de información de los datos de entrada reduciendo la redundancia y obteniendo las características más relevantes. Esta extracción se realiza normalmente mediante una interfaz de procesamiento de señal, pero las *features* extraídas por este método son muy generales y resulta difícil elegir cuales son útiles para una tarea concreta. Comúnmente para remediar este problema se utiliza la selección de *features*, que consiste en seleccionar los subconjuntos de *features* que minimizan el error. Otro enfoque conocido como aprendizaje de *features* realiza la extracción de *features* mediante aprendizaje automático, más concretamente utilizando redes neuronales profundas, intentando encontrar a su vez los más relevante, unificando el proceso en una sola fase. Para esta tarea en problemas basados en visión o audio se han utilizado redes convolucionales profundas debido a su capacidad de simulación de sentidos humanos (Choi et al., 2016).

## **1.4 Series temporales procedentes de narices artificiales**

Las narices artificiales son dispositivos diseñados para detectar distintos tipos de gases en el ambiente en el que se encuentran (Walt et al., 2012). Estos dispositivos se consideran no invasivos ya que no entran en contacto con el objeto de su estudio.

Para esta tarea se utilizan distintos tipos de sensores, siendo unos de los más utilizados los contruidos con semiconductores metal-óxido. Cuando son energizados, los electrones de estos semiconductores saltan a la banda de conducción, lo cual permite que sean atravesados por corrientes eléctricas. Esto se puede aprovechar teniendo en cuenta que distintos gases pueden causar reacciones redox, causando un cambio de conductividad en el semiconductor que puede ser medido (Ding et al., 2001; Zohora et al., 2013).

Los sensores de estos semiconductores no son invariantes en el tiempo, es decir, el valor de la conductividad depende no solo de las condiciones actuales, sino también de las pasadas, por lo que para un mismo gas la conductividad puede ser distinta en base a la situación pasada del sensor. Además, las condiciones climáticas (temperatura, humedad, etc.) también afectan a la eficacia de los sensores, ya que estos pueden deteriorarse debido a factores como la acumulación de suciedad o la agresión química. Todo esto hace que la utilización de las mediciones obtenidas directamente de los sensores para diferenciar distintos gases sea poco fiable.

Una forma de solucionar varios de estos problemas es limpiar el sensor entre gas y gas, devolviéndole a un estado neutro. Sin embargo, esto requiere de una monitorización continua del sensor, además de no ser válido para uso comercial.

Para abordar estos problemas, estos sensores suelen contar con resistores de calor para poder controlar la temperatura de la superficie del sensor, dotando a los gases de distintos valores de conductividad dependientes de la temperatura. Esta modulación además ayuda a que los sensores tengan una mayor capacidad de discriminación explotando distintas propiedades fisicoquímicas dependiendo de la temperatura. El cambio constante de temperatura es útil para reducir el efecto de los errores por histéresis, esto es, errores causados por la reluctancia del material a volver a su estado original tras recibir algún tipo de cambio.

Utilizando la modulación de temperatura se obtienen distintas medidas de conductividad de un sensor para distintas temperaturas y un mismo gas, obteniendo una representación de múltiples dimensiones (Yuan et al., 2020).

Otro de las características de las series producidas por estos sensores es la existencia de deriva en el sensor, por el cual los datos tomados para un mismo gas y temperatura pueden ser distinto para diferentes tiempos. El uso de aprendizaje automático es una de las formas de compensar este problema (Martinelli et al., 2013).

Uno de los enfoques utilizados para este tipo de tareas es la denominada como detección activa, que se basa en la modulación del sensor mediante información online del proceso olfativo, creando distintos perfiles de temperatura o cualquier otro parámetro modulable en base a la información obtenida por el sensor y especificándolos para un gas concreto. Esta información adicional se añade a la que aportan las series temporales multivariadas ya que la temporalidad de la modulación resulta relevante en los procesos de clasificación. Este

enfoque ha sido explorado para mejorar la eficiencia de los procesos de clasificación de series temporales provenientes de narices artificiales (Herrero-Carrón et al., 2015).

### ***1.5 Objetivos del TFG***

El objetivo de este TFG es estudiar el impacto de distintas estrategias de codificación y etiquetado en la clasificación de series temporales implementada con distintos algoritmos de clasificación y validarlas para diferentes series temporales, en concreto las procedentes de una nariz artificial para monitorizar de forma no invasiva rutinas humanas. En concreto, se analizarán series temporales que provienen de narices artificiales que monitorizan entornos educativos.

### ***1.6 Organización de la memoria***

La memoria consta de los siguientes capítulos:

- **Diseño y desarrollo**
- **Resultados**
- **Conclusiones y discusión**

## 2 Diseño y desarrollo

### 2.1 Estrategias de codificación

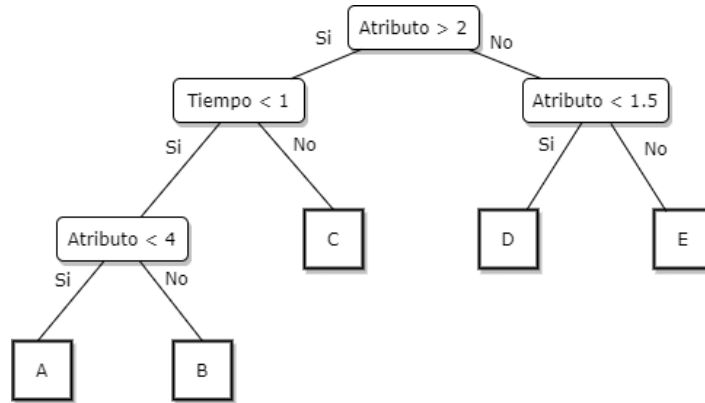
El formato que le vamos a dar a las series temporales para introducirlas en los distintos algoritmos encargados de procesarlas es muy importante, sobre todo debido a la necesidad de gestionar la manera en la que se hace uso de la temporalidad.

#### 2.1.1 Random Forest basado en SMTS

Un método de codificación más especializado del que haremos uso es la representación simbólica de series temporales en *Random Forest* (Baydogan and Runger, 2014). Para crear esta representación, cada dato de la serie original se codifica de la siguiente forma:

$$[t, x_1^n(t), x_1^n(t) - x_1^n(t - 1), x_2^n(t), \dots, x_m^n(t), x_m^n(t) - x_m^n(t - 1)]$$

Donde  $t$  denota el valor del tiempo en el que se recoge el dato con lo que conservamos la característica temporal de la serie y  $x_m^n(t) - x_m^n(t - 1)$  la diferencia entre el valor actual y el inmediatamente anterior del atributo  $m$  para los atributos numéricos. Esta diferencia es utilizada para extraer información de la tendencia de la serie y relacionarla con la clase a la que pertenece.



**Figura 2-1: Ejemplo de árbol de decisión del Random Forest para la codificación de series temporales.**

Con esta codificación se entrena un *Random Forest*  $RF_{ins}$  formado por  $J_{ins}$  árboles de decisión mediante las instancias individuales asignándole a cada una la clase de la serie a la que pertenecen. Este *Random Forest* consta de dos parámetros,  $J_{ins}$  que indica el número de árboles de decisión y  $R$  que denota el número de símbolos de nuestro alfabeto  $y$ , por tanto, número de nodos terminales del cada árbol del  $RF_{ins}$ . En la Figura 2-1 podemos ver un ejemplo de árbol con  $R = 5$ , formando un alfabeto de 5 símbolos. La elección de los valores de estos parámetros se explicará más adelante junto al algoritmo de clasificación que utiliza esta representación.

### 2.1.2 Algoritmos derivados de red convolucional

La forma más simple de introducir los datos a un algoritmo es alimentar la serie temporal sin ningún tipo de codificación. Esto no es recomendable en todos los algoritmos ya que muchos no son capaces de gestionar por sí mismos el orden de los datos y por consiguiente ignoran la temporalidad, pero hay algoritmos en los que esto sí es una posibilidad. Concretamente, los algoritmos derivados de redes convolucionales si son capaces de utilizar esta alimentación, ya que son redes que se aprovechan de la posición en la que se encuentra cada dato, por lo que pueden gestionar la temporalidad. Por tanto, la codificación de la alimentación es la siguiente para series multivariadas:

$$\begin{pmatrix} X_1^1 & X_1^2 & \dots & X_1^n \\ X_2^1 & X_2^2 & \dots & X_2^n \\ \dots & \dots & \dots & \dots \\ X_m^1 & X_m^2 & \dots & X_m^n \end{pmatrix}$$

Y la siguiente para series monovariadas:

$$(X_1 \ X_2 \ \dots \ X_n)$$

Donde el superíndice denota el atributo y el subíndice el instante de tiempo.

## 2.2 Estrategias de etiquetado

Para poder llevar a cabo el entrenamiento de los distintos clasificadores, necesitamos etiquetar los datos de los que disponemos según sus características. Por ejemplo, en el caso de la clasificación de señales olfativas de rutinas en entornos educativos, podemos etiquetar los datos con la asignatura a la que pertenecen, como por ejemplo si corresponden a una hora de clase de matemáticas. En este caso el etiquetado se puede realizar basándonos en el tiempo en el que ha sido tomado cada dato y considerando el horario escolar correspondiente a las distintas clases. Aunque es posible realizar esta tarea manualmente, el volumen de datos disponible sugiere la necesidad de automatizar el proceso.

Mientras que en algunos casos la correcta automatización del proceso de etiquetado puede llegar a requerir de aprendizaje automático, la automatización en nuestro caso es bastante simple ya que nos basamos únicamente en el tiempo en el que fueron recogidos los datos. Teniendo en cuenta que los datos de los que disponemos provienen de una clase de primaria, podemos realizar el etiquetado de series en base a distintos factores.

Uno de los enfoques más simple es la distinción de días lectivos de no lectivos. Las series temporales estarán compuestas de un número de datos correspondiente a un día entero, y la etiqueta se decide en base a si el día forma parte de la semana lectiva (de lunes a viernes) y teniendo en cuenta los días festivos. Este etiquetado establece el problema de clasificación más simple que se puede abordar con nuestros datos, ya que la actividad humana se reduce a cero en fines de semana y días festivos.

De manera más concreta, podemos etiquetar como lectiva o no cada franja horaria de 45 minutos propia al tiempo de cada clase. El correspondiente problema de clasificación es ligeramente más complejo ya que después de una clase los niveles de las distintas medidas disminuyen lentamente en vez de ser totalmente nulos en los sensores.

Otro etiquetado aún más simple es la división en días de la semana. Es decir, especificar que la actividad registrada en los sensores corresponde al lunes, al martes, etc. Sin embargo, el problema de clasificación que crea es más complejo ya que todos los días lectivos no tienen la misma franja horaria en la que existe actividad humana.

El nivel de atención de los alumnos en clase es un factor que puede ser medido mediante los niveles de  $\text{CO}_2$  ya que el consumo de este gas aumenta con la atención cognitiva. Sin embargo, no podemos etiquetar este hecho directamente ya que no disponemos de información adicional del nivel de atención (por ejemplo, registrado con alguna anotación del profesorado). Sin embargo, esta variación de  $\text{CO}_2$  puede ser utilizada para intentar diferenciar entre distintas asignaturas ya que es muy posible que existan diferencias entre los niveles de atención entre una asignatura y otra. El etiquetado para esta tarea consistirá en utilizar el horario proporcionado por el colegio para dividir los datos en series temporales de 45 minutos cada una correspondientes a cada clase, siendo la etiqueta el nombre de la asignatura.

La realización de exámenes es otro evento notable que puede ser etiquetado. Sin embargo, debido a la baja cantidad de exámenes de los que disponemos, no es un problema de clasificación abordable.

También etiquetaremos las series temporales en base a si son días peculiares. Esto es, días en los que en algún momento no se siguió el horario de manera habitual, como puede ser una excursión, o una charla. Estos días fueron confirmados por el centro.

Por último, y con la diferencia de que los datos que se usaran para esto provienen de aulas de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid y no del colegio, etiquetaremos una serie de horas correspondientes a otra rutina humana en las que se realizó trabajo de limpieza, y por tanto se podría esperar emisiones de distintas sustancias a mayores niveles que en cualquier otra hora.

## **2.3 Algoritmos de clasificación empleados en este TFG**

### **2.3.1 Random Forest basado en SMTS**

En base a la representación simbólica mencionada anteriormente en las estrategias de codificación, se hará uso de un conjunto de  $J_{ts}$  árboles de decisión para la clasificación de las series en base a la frecuencia de cada símbolo del alfabeto obtenido en la codificación (Baydogan and Runger, 2014). Denotaremos el *Random Forest* formado por estos árboles como  $\text{RF}_{ts}$ . Este enfoque de clasificación se conoce como bolsa de palabras, siendo las palabras los símbolos de nuestro alfabeto.

Incluyendo la codificación y creación del alfabeto, este algoritmo requiere de tres parámetros.  $R$  denota el número de símbolos en nuestro alfabeto que se corresponde al



número de nodos terminales de cada árbol de  $RF_{ins}$ . Este número de árboles viene dado por el parámetro  $J_{ins}$ , mientras que el número de árboles del  $RF_{ts}$  se denota con  $J_{ts}$ .

El parámetro  $R$  determina el nivel de detalle que se obtiene de la representación simbólica creada por  $RF_{ins}$ . Un método para elegir este valor es la evaluación de los valores de  $R$  para un valor de  $J_{ins}$  fijo. Tras obtener las representaciones simbólicas se entrena  $RF_{ts}$  con cada una de ellas con  $J_{ts}$  fijo, y se selecciona el valor de  $R$  que obtenga la mejor ratio de error OOB (Out of bag). Utilizando este valor, obtenemos la representación simbólica en base a los distintos valores de  $J_{ins}$ , y de nuevo para cada una de ellas se entrena  $RF_{ts}$  con  $J_{ts}$  fijo, obteniendo de igual manera el  $J_{ins}$  cuyo error OOB sea el mejor. Utilizando de nuevo la evaluación mediante OOB por última vez, seleccionamos el mejor valor de  $J_{ts}$  con incremento discreto usando para los otros dos parámetros los valores obtenidos anteriormente. Fijando una tolerancia, el valor de  $J_{ts}$  se escoge cuando la mejora de su OOB en comparación con el incremento anterior es menor de una tolerancia fija.

### 2.3.2 Fully convolutional network (FCN)

Otro algoritmo que se va a utilizar en este trabajo es una red completamente convolucional (FCN) que ha mostrado buenos resultados con respecto a la clasificación de series temporales dentro de la familia de algoritmos de aprendizaje profundo (Wan et al., 2019). Las FCN son redes convolucionales que no contienen ninguna capa de *pooling* local, manteniendo así las dimensiones originales de la entrada durante todo el proceso hasta la última capa, que en vez de ser completamente conexa se sustituye por una capa de *average pooling* global que reduce el número de parámetros requeridos. Debido a que la posición de cada atributo es relevante en las redes convolucionales, las series completas pueden ser introducidas sin perder su temporalidad.

La arquitectura que se va a utilizar (Wang et al., 2017) se muestra en la Figura 2-2 y consta de tres bloques convolucionales, cada uno de ellos compuesto por una capa convolucional, utilizando normalización por lotes y función de activación ReLu. El último bloque está conectado a la capa de *average pooling* global cuya salida a su vez está conectada a un clasificador *softmax* del cual se obtiene la clase. Las dimensiones de la capa de entrada son las correspondiente a la codificación base, numero de atributos por número de pasos de tiempo. Las capas ocultas tienen como dimensiones el número de pasos de tiempo por el número de filtros de cada una, 128, 256 y 128 respectivamente. El tamaño de los *kernels* (filtros) es de 3,5 y 8. Esta arquitectura ha sido ya utilizada anteriormente en clasificación de series temporales, obteniendo resultados vanguardistas.

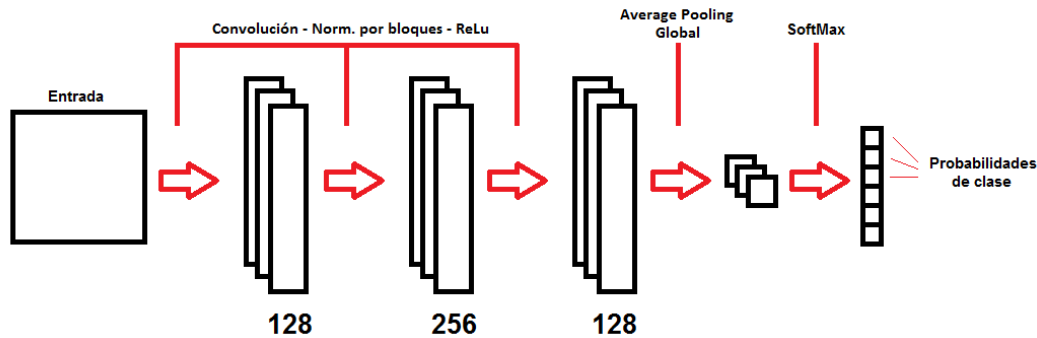
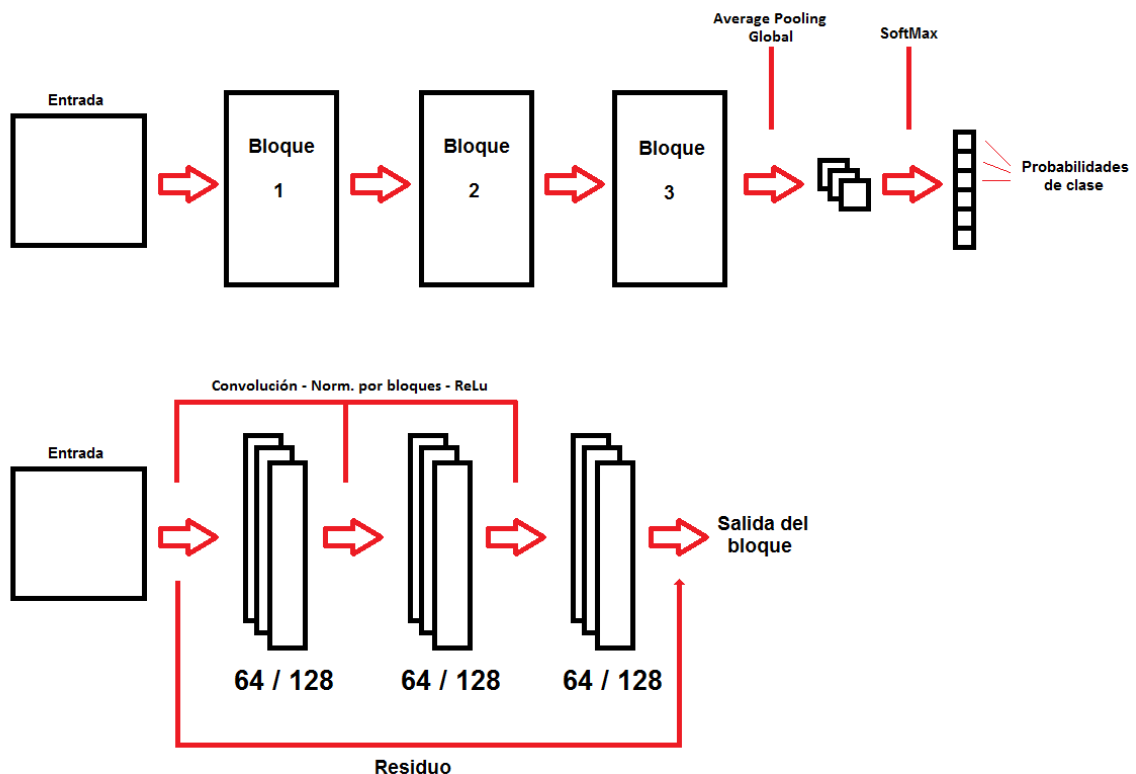


Figura 2-2: Arquitectura de FCN utilizada en este trabajo.

### 2.3.3 Red profunda residual (RNN)

El tercer algoritmo que vamos a utilizar es una arquitectura de red residual. Como se mencionó anteriormente, las redes residuales son algoritmos de aprendizaje profundo que además de utilizar la convolución, tienen conexiones residuales entre bloques consecutivos de capas para reducir la problemática del desvanecimiento de gradiente, lo que puede llegar a frenar con demasiada rapidez el proceso de aprendizaje. Estas redes obtienen muy buenos resultados en clasificación de series temporales, destacando sobre el resto de los algoritmos de aprendizaje profundo en series monovariadas y asemejándose a los resultados de las FCN en series multivariadas (Wan et al., 2019).

La arquitectura que vamos a utilizar (Wang et al., 2017) se muestra en la Figura 2-3 y consta de tres bloques residuales todos compuestos de tres capas convolucionales de 64 filtros cada una para el primer bloque y 128 para los otros dos a cuya salida se le añade la entrada del bloque debido a la conexión residual. Al igual que en la FCN utilizada, a cada convolución le sigue la normalización por lotes y función de activación ReLu. Estos bloques preceden a una capa de *average pooling* global y la red termina con un clasificador *softmax* de donde se obtiene la clase de la serie. Al igual que con la arquitectura de la FCN utilizada, esta arquitectura de RNN ha obtenido resultados vanguardistas en cuanto a clasificación de series temporales se refiere.



**Figura 2-3: Arquitectura de red residual utilizada. Panel superior: arquitectura de red. Panel inferior: arquitectura de bloque. Se aplican 64 filtros para el bloque 1 y 128 para los bloques 2 y 3.**

## 2.4 Descripción de las series temporales de la nariz artificial

Las series temporales provenientes de las narices artificiales constan de un total de 19 atributos correspondientes al número de medidas captadas por los sensores de la nariz, y del tiempo (*timestamp*) correspondiente al momento en el que fueron captadas. El periodo entre cada captura es de 1 minuto, obteniendo así 1440 datos por cada día en el que la nariz estaba funcionando. Los datos utilizados pertenecen al periodo lectivo comprendido entre el 16 de septiembre y el 20 de diciembre de 2019.

Sensor	Descripción del contenido de la señal
IR (infrarrojos)	Nivel de radiación electromagnética infrarroja
UV (Ultravioleta)	Nivel de radiación ultravioleta
bme280_humidity	Nivel de humedad en el ambiente
bme280_pressure	Nivel de presión
bme280_temperature	Nivel de temperatura
co2_digital	Nivel de CO <sub>2</sub>
door_is_open	1 cuando la puerta está abierta, 0 cuando está cerrada
loudness	Nivel de sonido
luminosity	Nivel de iluminación de la zona.
pm10	Nivel de partículas de menos de 10 micras
pm25	Nivel de partículas de menos de 2,5 micras
tgs2600	Nivel de contaminantes del aire. Hidrógeno y CO
tgs2602	Nivel de contaminantes del aire. Amoníaco y H <sub>2</sub> S
tgs2603	Nivel de aminas
tgs2610	Nivel de gas LP (Licuado del petróleo)
tgs2611	Nivel de metano (CH <sub>4</sub> )
tgs2620	Vapores orgánicos
vbatt	Nivel de batería
wiffi_rissi	Señal de wifi

Tabla 2-1: Atributos de las series temporales

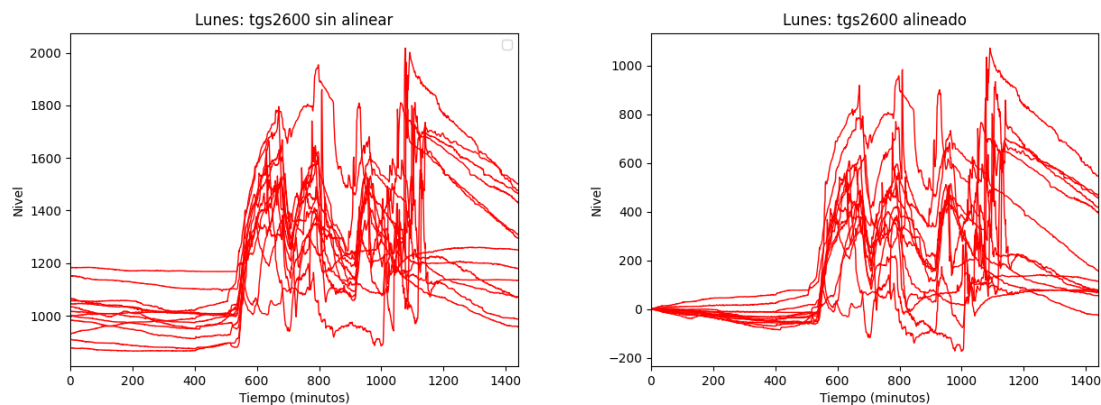
De entre estas medidas, las que más nos interesan son el nivel de CO<sub>2</sub>, la humedad, y las que reflejan las distintas sustancias orgánicas en el aire, ya que todas ellas están altamente relacionadas con la actividad humana. Por otro lado, los niveles de batería y señal de wifi no serán utilizados en este proyecto, aunque sí que pueden verse ligeramente afectados por el número de personas en el aula y podrían considerarse en estudios futuros donde este dato sea relevante. Una de las ventajas de la recogida de datos en el colegio es que el número de personas permanece constante. Las medidas relacionadas con la luminosidad como puede ser la radiación también podrían ser relevantes en otros proyectos donde no se conocieran bien los horarios y el encendido y apagado de luces podría utilizarse para etiquetar los datos.

## 2.5 Metodología de análisis

Para la realización de pruebas vamos a clasificar los distintos conjuntos de datos que hemos creado en base al etiquetado descrito en la sección 2.2.

Para su introducción en el *Random Forest*, se transforma cada serie a su representación simbólica, mientras que para FCN y RNN no se realiza ningún tipo de modificación. Los datos utilizados han sido recogidos de un total de 14 semanas previas al estado de pandemia, empezando desde el lunes 16 de septiembre y acabando el domingo 21 de diciembre de 2019. De este periodo se ha tenido en cuenta que han existido 4 días entre semana no lectivos.

En el caso de los atributos correspondientes a los sensores TGS, utilizaremos alineación en los problemas de días completos. Como vemos en la Figura 2-4, la alineación consiste en fijar al mismo valor el comienzo de todas las series eliminando el offset inicial de las señales, para evitar problemas con valores residuales de otros días. Esta metodología asume la hipótesis de que la evolución temporal en los sensores no se ve muy afectada por un ligero offset constante en la amplitud de las señales.



**Figura 2-4: Ejemplo de alineación de señales en los sensores (TGS2600)**

Debido a la reducida cantidad de datos de los que disponemos y para mayor exhaustividad en las pruebas, haremos uso de la validación cruzada. Esto es, para un conjunto de datos dado, realizaremos una división en 5 grupos del mismo tamaño de los cuales 4 formaran parte de la partición de entrenamiento y el restante se utilizará para la validación, realizando un total de 5 veces las pruebas de clasificación, una por cada grupo. Las particiones de

entrenamiento y validación por lo tanto tendrán un 80% y un 20% de los datos respectivamente.

En el caso del conjunto de datos de días peculiares, debido a que disponemos de un número especialmente reducido de ellos, hemos creado una sola división entre entrenamiento y validación a mano en base al conjunto de datos de días lectivos y no lectivos.

Para los problemas en los que existen más de dos clases, observaremos para cada clase su precisión y sensibilidad (*recall*) propios para analizar si el acierto/fallo de alguno de los problemas se centra en alguna de las clases. La precisión corresponde al número de elementos identificados correctamente como positivos de entre todos los identificados como positivos, dando importancia a que los casos reconocidos como positivos lo sean. Por otro lado, la sensibilidad corresponde al número de elementos identificados como positivos de entre todos los elementos positivos, buscando por tanto identificar todos los casos positivos.



### 3 Resultados

A continuación, se exponen los resultados de las distintas pruebas para los problemas de clasificación mencionados anteriormente. Para cada problema se mostrará una tabla que contendrá el % de acierto de cada clasificador para las particiones de entrenamiento y validación de la mejor combinación de atributos encontrada y de cada atributo individual.

La primera labor para alimentar los datos a los clasificadores consistió en asignar etiquetas a las distintas series temporales correspondientes a horas o días en función de las distintas características de su franja temporal. Es decir, en el etiquetado utilizamos la información del día, la semana, el horario de inicio de la actividad y el tipo de actividad a la que corresponde cada serie. De esta manera se ha podido segmentar los datos para el entrenamiento y asignarles su correspondiente etiqueta. Con cada tipo de etiquetado, creamos un problema de clasificación distinto con el que se han realizado las pruebas a partir de los datos disponibles y que se comentarán en cada sección de este capítulo.

Los resultados se han obtenido mediante los tres algoritmos distintos mencionados anteriormente: *Random Forest* basado en SMTS, red completamente convolucional, y red profunda residual. Los parámetros para cada uno de ellos se muestran en la Tabla 3-1, Tabla 3-2 y Tabla 3-3 respectivamente.

Parámetro	Valor
R (Símbolos del alfabeto)	$\in \{20, 50, 100, 250, 500\}$
$J_{ins}$ (Número de árboles del <i>Random Forest</i> de codificación)	$\in \{20, 50, 100, 250, 500\}$
Incremento para $J_{ts}$ (Número de árboles del <i>Random Forest</i> de clasificación)	50
T (Diferencia temporal para la codificación)	1

Tabla 3-1: Parámetros Random Forest basado en SMTS.

Parámetro	Valor
Bloques convolucionales	3
Estructura por bloque	Capa convolucional Normalización en lote Función de activación: ReLU
Número de filtros por bloque	128, 256, 128
<i>Kernels</i> por convolución	3
Tamaños <i>kernels</i>	8, 5, 3
Épocas	500
Tamaño de lote	64
<i>Pooling</i>	<i>Global Average Pooling</i>
Función de salida	<i>Softmax</i>

Tabla 3-2: Parámetros FCN.

Parámetro	Valor
Bloques residuales	3
Estructura por bloque	Capa convolucional 1 Normalización en lote Función de activación: ReLU Capa convolucional 2 Normalización en lote Función de activación: ReLU Capa convolucional 3 Normalización en lote Función de activación: ReLU
Número de filtros por bloque	{64, 64, 64}, {128, 128, 128}, {128, 128, 128}
<i>Kernels</i> por convolución	3
Tamaños <i>kernels</i>	{8, 5, 3} Por cada bloque
Épocas	500
Tamaño de lote	64
<i>Pooling</i>	<i>Global Average Pooling</i>
Función de salida	<i>Softmax</i>

Tabla 3-3: Parámetros RNN

### 3.1 Problema de días lectivos

Para la distinción entre días lectivos y no lectivos disponemos de un conjunto de datos de 98 series temporales con 1440 datos por cada una correspondientes al número de minutos de cada día de las 14 semanas que se han utilizado, siendo los días no lectivos fines de semana y festivos y lectivos el resto, por tanto, el problema consta de dos clases. Estas 98 series se han dividido en 77 para entrenamiento y 21 para validación.

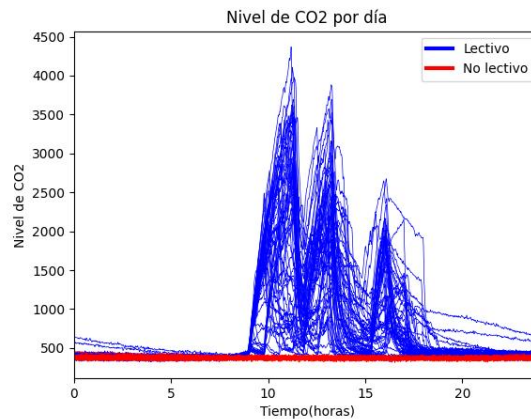
	SMTS	FCN	RNN
Todos	97.73	100	100
CO <sub>2</sub>	100	100	100
Humedad	97,6	99,0	100
TGS2600	99,8	99,0	100
TGS2602	99,1	100	100
TGS2603	99,8	100	100
TGS2610	98,9	98,0	100
TGS2611	99,1	98,0	100
TGS2620	100	100	100

Tabla 3-4: % de acierto problema de días lectivos

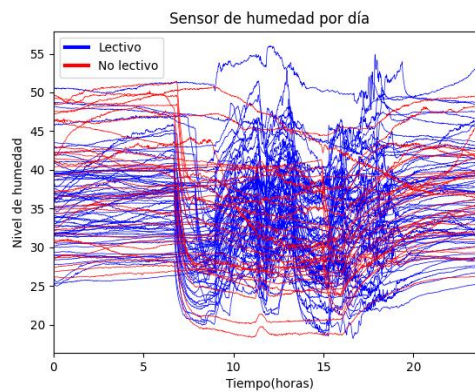
Como podemos observar en Tabla 3-4: % de acierto problema de días lectivos, se obtienen resultados muy positivos en todas las combinaciones, aunque debido a la simplicidad del



problema era de esperar encontrarse con buenos resultados. Es destacable que el uso de CO<sub>2</sub> y TGS2620 individualmente obtiene siempre un 100% de acierto, pero el resto de las configuraciones no se quedan atrás, no bajando ninguna de un 97% de acierto.



**Figura 3-1: Nivel de CO<sub>2</sub> durante cada día de curso, divididos en lectivos y no lectivos**



**Figura 3-2: Nivel de humedad durante cada día de curso, divididos en lectivos y no lectivos**

Si nos fijamos en la Figura 3-1, podemos observar que el nivel de CO<sub>2</sub> para los días lectivos es muy superior y bastante más variable que el de los no lectivos. Por otro lado, los niveles de humedad mostrados en la Figura 3-2 presentan mucha menos claridad en su diferenciación, pero la variabilidad capturada es suficiente para que los clasificadores realicen una clasificación acertada.

### **3.2 Problema de periodos lectivos**

Recordamos que en este problema vamos a clasificar franjas de 45 minutos entre lectivas y no lectivas. Para equilibrar el número de series de cada clase, escogemos por cada día en el que haya clase las 6 franjas lectivas y como franjas no lectivas dos de antes del comienzo de las clases, dos después del final, y dos en el descanso de la comida. Estas franjas se han elegido ya que se encuentran cerca de horas lectivas, con lo que tienen valores residuales en las mediciones, sin lo cual el problema es tan simple como el anterior. Con esto obtenemos un total de 792 series temporales con 2 clases posibles, agrupadas en 632 para entrenamiento y 160 para validación.

	SMTS	FCN	RNN
Todos	88,6	87,1	85,0
CO <sub>2</sub>	89,7	85,2	86,9
Humedad	78,4	76,6	74,5
TGS2600	83,1	80,7	80,5
TGS2602	79,0	75,1	77,2
TGS2603	84,2	80,54	79,7
TGS2610	82,9	81,0	80,1
TGS2611	83,3	81,4	80,1
TGS2620	82,8	80,4	79,9

**Tabla 3-5: % de acierto problema de periodos lectivos**

Podemos observar que se obtienen tasas de acierto elevadas, destacando el uso del nivel de CO<sub>2</sub> seguido por el uso de todas las variables, y a continuación el sensor TGS2603 (Aminas). Dicho esto, podrían esperarse resultados mejores ya que como vimos en la Figura 3-1 la diferencia entre el nivel del CO<sub>2</sub> y otras medidas entre la franja escolar y el resto del día es muy notable. Viendo esto, la principal hipótesis que manejamos es que gran parte de las horas de educación física y música se realizan fuera del aula, creando probablemente niveles residuales similares a las franjas después de clase y en la franja de la comida.

En base a todo esto, decidimos crear un segundo conjunto de datos sin introducir las series correspondientes a las dos asignaturas mencionadas, eliminando de cada día un número de periodos no lectivos igual al número de ocurrencias de dichas clases para compensar el número de series de cada clase, reduciendo el total a 672 series divididas en 536 para entrenamiento y 136 para validación.

	SMTS	FCN	RNN
Todos	91,8	91,0	91,7
CO <sub>2</sub>	92,0	90,7	91,0
Humedad	81,7	81,2	79,3
TGS2600	86,7	84,3	84,9
TGS2602	82,6	81,4	80,4
TGS2603	87,7	86,4	84,6
TGS2610	87,2	85,3	83,0
TGS2611	87,2	86,8	84,9
TGS2620	86,7	85,7	84,1

**Tabla 3-6: % de acierto problema de periodos lectivos reducido**

Podemos notar una clara mejoría con respecto a los resultados anteriores, llegando a alcanzar un 92% de acierto en la clasificación teniendo en cuenta solo el CO<sub>2</sub>. Esto denota que efectivamente parte de los errores de clasificación se debían a las asignaturas que normalmente no se realizaban en el aula.

Sin embargo, aún con este incremento, podríamos esperar una precisión incluso mayor, ya que como hemos podido observar la diferencia de medidas relativas a la actividad humana en las aulas es muy notable. La razón por la que siguen existiendo errores de clasificación es debido a peculiaridades de días concretos como por ejemplo una muy baja asistencia, o sobre todo la ausencia total de personas por alguna actividad extraordinaria como puede ser una excursión o algún tipo de charla que no hubiese constado como tal.

Al igual que la clasificación de días lectivos, este problema es bastante simple y se puede realizar una clasificación muy precisa utilizando únicamente el CO<sub>2</sub>, aunque hay otras medidas que también reportan buenos resultados.

### **3.3 Problema de días de la semana**

Para la clasificación de los días de la semana disponemos de 66 series temporales ya que hemos decidido no tener en cuenta los días festivos y fines de semana debido a la similitud entre estos días por falta de actividad humana. Por lo tanto, las series se dividen en 5 clases distintas correspondientes a lunes, martes, miércoles, jueves y viernes. La partición utilizada es de 51 series para entrenamiento y 15 para validación.

	SMTS	FCN	RNN
Todos	25,4	37,4	28,7
CO <sub>2</sub>	55,2	50,1	46,6
Humedad	47,1	31,6	30,4
TGS2600	50,4	34,9	36,3
TGS2602	34,1	25,9	33,4
TGS2603	41,7	26,6	36,3
TGS2610	44,4	38,1	37,6
TGS2611	41,6	34,9	42,2
TGS2620	44,0	35,0	40,8

**Tabla 3-7: % de acierto problema de días de la semana**

Los resultados obtenidos mostrados en la Tabla 3-7 no son excesivamente favorables. La combinación del conjunto de datos CO<sub>2</sub> con el uso de SMTS reporta un acierto destacable pero aun así ligeramente bajo. Esto se puede deber a diversas actividades no planificadas que alteran el uso del aula durante el día y afectan así a los niveles de las distintas medidas. Por otro lado, el calendario, adjuntado en el anexo A, contempla una distribución variada de asignaturas entre días de la semana.

	Precisión	Sensibilidad
Lunes	57,3	69,2
Martes	73,3	62,8
Miércoles	37,5	42,8
Jueves	69,8	73,0
Viernes	35,6	25,8

**Tabla 3-8: Precisión y sensibilidad (en %) de cada día de la semana para SMTS utilizando únicamente los niveles de CO<sub>2</sub>**

Como podemos observar en la Tabla 3-8, el jueves y el martes son los días con mejores resultados seguidos por el lunes, obteniendo los tres días precisión y sensibilidad superiores al 60%. Tanto el miércoles como el viernes obtienen menos de un 50% en ambas medidas, denotando mayor dificultad de clasificación en ellos.

### **3.4 Problema de asignaturas**

El problema de clasificación de asignaturas consta de 396 series temporales de 45 datos cada una, que al igual que en el problema de periodos lectivos corresponden a las franjas de 45 minutos en las que se imparte una asignatura, indicadas en el calendario provisto por el centro, disponible en el anexo A. El total de asignaturas y por tanto clases es de 8. Las series se dividen en 315 para entrenamiento y 81 para validación.

	SMTS	FCN	RNN
Todos	27,5	28,5	28,2
CO <sub>2</sub>	31,2	30,1	30,7
Humedad	27,9	30,1	27,0
TGS2600	27,7	28,2	28,5
TGS2602	23,5	27,5	24,4
TGS2603	30,1	31,0	29,0
TGS2610	27,2	29,0	25,4
TGS2611	28,2	30,3	29,0
TGS2620	28,7	30,8	22,7

**Tabla 3-9: % de acierto problema de asignaturas**

Se puede observar en la Tabla 3-9 que los resultados son bajos en todos los casos. De entre ellos podemos destacar la clasificación por niveles de CO<sub>2</sub> y por el sensor TGS2603 (series de aminos, olores sulfurosos) sin tener ninguno de los clasificadores una ventaja clara frente a los demás.

	Precisión	Sensibilidad
Lengua	27,9	36,5
Matemáticas	32,8	51,1
Ciencias naturales	32,0	18,9
Ciencias sociales	22,7	6,6
Música	34,8	27,7
Educación Física	50,0	58,9
Inglés	16,8	12,1
Valores	5,5	0,7

**Tabla 3-10: Precisión y sensibilidad (en %) de cada asignatura para SMTS utilizando únicamente los niveles de CO<sub>2</sub>**

En la Tabla 3-10 observamos que la clase de valores es, de lejos, la que más difícil resulta clasificar. Educación física es la que obtiene mejores resultados, ya que es la única que siempre se realiza fuera del aula. El resto de las asignaturas obtienen resultados poco destacables.

Viendo esto, para concretar el problema, clasificamos ahora las series de dos asignaturas. En concreto, vamos a comparar la asignatura de lengua con otras tres.

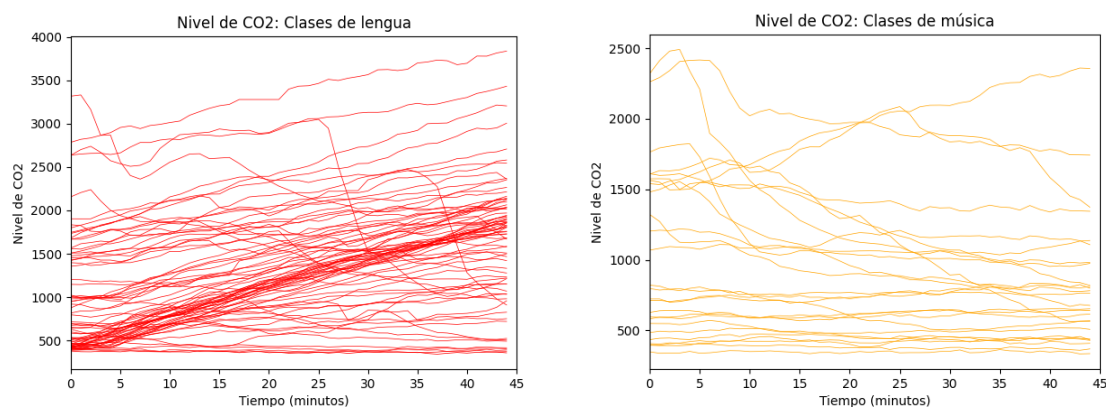
Empezamos comparando con la asignatura de música, la cual solo se lleva a cabo en la clase en la que se encuentra el sensor en contadas ocasiones. Contamos con 105 series temporales, que se dividen en 85 para entrenamiento y 20 para validación. Hay que tener en cuenta que tenemos tres veces más series de lengua que de música.

	SMTS	FCN	RNN
Todos	84,8	86,6	84,5
CO <sub>2</sub>	85,2	87,6	92,3
Humedad	79,1	87,6	86,6
TGS2600	82,0	89,5	92,3
TGS2602	80,1	80,9	79,9
TGS2603	79,1	85,7	88,5
TGS2610	83,1	86,7	87,5
TGS2611	79,0	86,6	89,5
TGS2620	82,8	89,5	86,6

**Tabla 3-11: % de acierto problema de asignaturas (Solo lengua y música)**

En la Tabla 3-11 se observa que, aun teniendo en cuenta la diferencia entre número de series de cada clase, se obtienen resultados positivos, llegando al 91,5. En la Figura 3-3 podemos ver las medidas de CO<sub>2</sub> de ambas asignaturas, donde se observa la clara diferencia entre ellas. Los fallos de clasificación se pueden deber principalmente a los valores bajos o descendentes en las clases de lengua, propiciados por situaciones poco comunes que no

corresponden a un periodo normal, y a las clases de música que, si se llevaron a cabo en el aula, siendo esto una situación inusual.



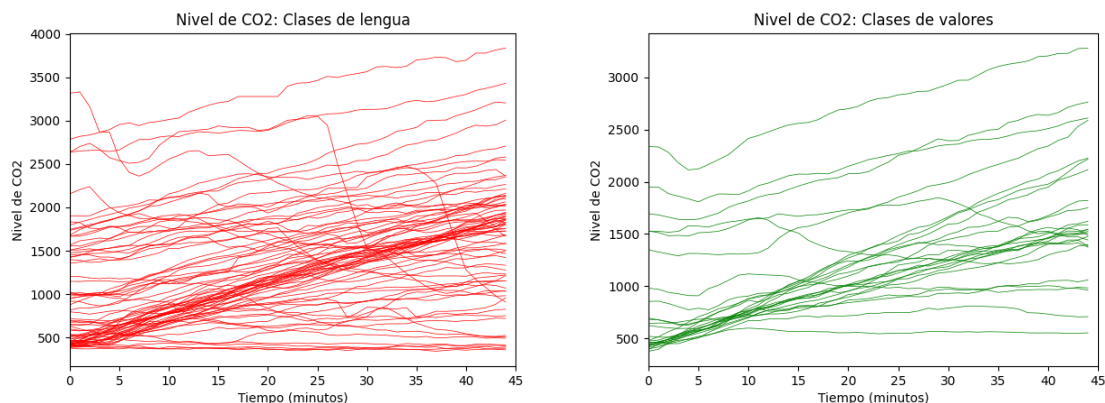
**Figura 3-3: Gráficas de CO<sub>2</sub> en clases de lengua y música**

A continuación, comparamos con la asignatura de valores, la cual al igual que lengua se realiza siempre en el aula, pero en la que podríamos esperar un nivel de atención distinto, lo cual se mostraría sobre todo en el CO<sub>2</sub>. El conjunto de datos usado está formado por 105 series, divididas en 83 para entrenamiento y 22 para validación. De nuevo, hay diferencia entre el número de series de cada asignatura.

	SMTS	FCN	RNN
Todos	73,1	79,9	82,8
CO <sub>2</sub>	72,7	76,14	78,0
Humedad	71,1	75,2	77,1
TGS2600	75,1	80,9	82,8
TGS2602	74,5	77,1	76,1
TGS2603	77,2	80,9	84,7
TGS2610	76,6	80,8	81,8
TGS2611	73,7	82,8	77,1
TGS2620	76,9	80,9	81,8

**Tabla 3-12: % de acierto problema de asignaturas (Solo lengua y valores)**

Aunque en la Tabla 3-12 pueda parecer que los resultados son mayormente positivos, hay que tener en cuenta la diferencia entre series de cada clase. En general los clasificadores no consiguen aprender con una precisión demasiado alta. En la Figura 3-4 se puede observar que la tendencia de ambas asignaturas es similar, situando la mayoría de las series en una trayectoria ascendente desde 500 a 2000.



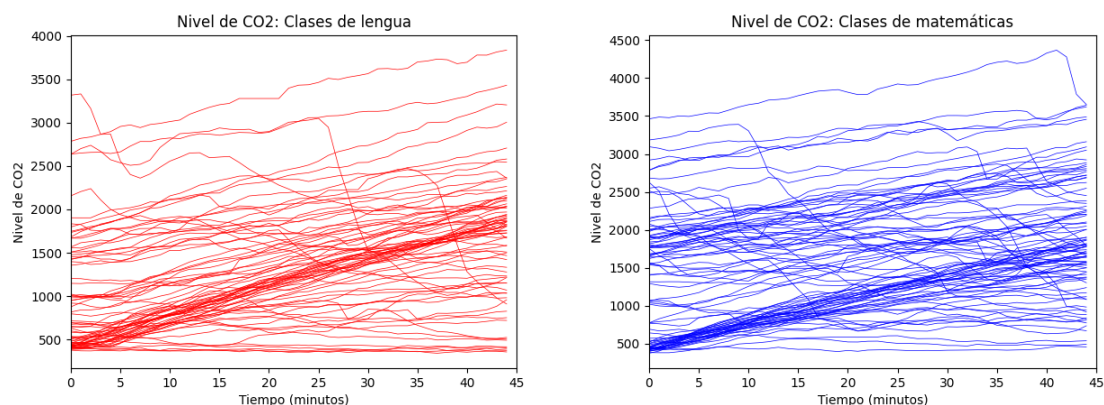
**Figura 3-4: Gráficas de CO<sub>2</sub> en clases de lengua y valores**

Por último, comparamos la clase de lengua con la clase de matemáticas. Ambas clases se realizan siempre en el aula, tienen casi el mismo número de horas semanales, y al ser las dos asignaturas "principales" podríamos esperar un nivel de atención similar. El conjunto de datos en este caso está formado por 171 series, divididas en 137 para entrenamiento y 34 para validación.

	SMTS	FCN	RNN
Todos	62,6	69,5	70,1
CO <sub>2</sub>	63,7	62,0	63,1
Humedad	59,3	65,4	69,6
TGS2600	62,2	68,3	68,4
TGS2602	55,7	62,54	62,5
TGS2603	62,2	64,3	64,2
TGS2610	59,4	70,1	68,9
TGS2611	60,1	68,9	69,6
TGS2620	63,3	71,2	67,2

**Tabla 3-13: % de acierto problema de asignaturas (Solo lengua y matemáticas)**

Viendo la Tabla 3-13, los resultados, aunque positivos, dejan margen de mejora, no alcanzando el 70% de acierto en ninguna de las combinaciones. Si observamos la Figura 3-5, podemos observar que visualmente las clases no presentan ninguna diferencia clara entre sí. Pese a esto, los clasificadores son capaces de acertar una mayor parte de las series.



**Figura 3-5: Gráficas de CO<sub>2</sub> en clases de lengua y matemáticas**

En base a esto, a priori el problema de clasificación no tiene solución simple. El 30% de acierto que se alcanza se debe probablemente a diferencias claras con las asignaturas de música y educación física, ya que son las únicas que tienen clases fuera del aula, y aún en esos casos pueden encontrarse suficientes similitudes entre las dos como para confundir la una con la otra. Aun con todo esto, los análisis individuales han mostrado que los clasificadores pueden clasificar correctamente un mayor número de elementos de los que se podrían identificar visualmente.

### 3.5 Problema de franjas horarias

En este problema queremos clasificar cada franja horaria en la hora lectiva a la que pertenece, es decir, 1<sup>ra</sup> hora, 2<sup>da</sup>, etc... teniendo un total de 6 clases. El número de series es 396, divididas en 312 para entrenamiento y 84 para validación, estando balanceado el número de series de cada clase.

	SMTS	FCN	RNN
Todos	47,0	47,1	43,1
CO <sub>2</sub>	52,6	51,7	49,4
Humedad	27,5	33,2	30,5
TGS2600	37,7	31,5	34,2
TGS2602	37,1	32,5	29,0
TGS2603	44,2	43,8	43,8
TGS2610	36,8	36,3	31,3
TGS2611	40,5	41,1	36,8
TGS2620	38,8	35,5	31,7

**Tabla 3-14: % de acierto problema de franjas horarias**

En la Tabla 3-14 podemos observar que los resultados obtenidos son ligeramente positivos, teniendo en cuenta que si se clasifica de forma aleatoria entre 6 clases se obtendría un 16.6% de acierto de media. La clasificación por nivel de CO<sub>2</sub> vuelve a destacar notablemente frente al resto de medidas seguido por el uso de todas las medidas y el sensor TGS2603 (series de



aminas, olores sulfurosos), mientras que el *Random Forest* basado en SMTS obtiene resultados ligeramente superiores a los otros dos algoritmos. A continuación, observaremos los resultados para cada franja individualmente.

	Precisión	Sensibilidad
1 <sup>ra</sup>	87,1	90,4
2 <sup>nda</sup>	42,5	39,6
3 <sup>ra</sup>	41,3	36,9
4 <sup>ta</sup>	51,0	54,3
5 <sup>ta</sup>	34,2	27,8
6 <sup>ta</sup>	52,6	66,0

**Tabla 3-15: Precisión y sensibilidad (en %) de cada franja horaria para SMTS utilizando únicamente los niveles de CO<sub>2</sub>**

Como observamos en la Tabla 3-15, la primera franja horaria es esperablemente más fácil de identificar ya que siempre parte de actividad nula. La cuarta y sexta franjas destacan sobre las otras tres franjas ya que corresponden a la vuelta del recreo y la hora de comer respectivamente, que, aunque no llegan a partir de valores base, se ven afectado por estos periodos de baja actividad.

### 3.6 Problema de días peculiares

En este problema intentamos clasificar días que se salen de las rutinas habituales, como puede ser una excursión, o una charla. El colegio informó de dicha peculiaridad. En comparación con el resto de los problemas, en este realizamos validación simple con las particiones creadas manualmente debido a la baja cantidad de ejemplos. La partición de entrenamiento consta de 51 días lectivos y 28 no lectivos, mientras que la de validación tiene 10 días lectivos, 5 días no lectivos y los 5 días peculiares de los que disponemos, perteneciendo los días lectivos a una clase y los no lectivos y peculiares a la otra. En este problema buscamos si es posible la generalización para poder diferenciar días lectivos normales de peculiares.

	SMTS	FCN	RNN
Todos	75,5	100	100
CO <sub>2</sub>	75,0	100	100
Humedad	75,0	95,0	90,0
TGS2600	75,0	90,0	85,0
TGS2602	75,0	90,0	85,0
TGS2603	75,0	95,0	90,0
TGS2610	77,5	75,0	75,0
TGS2611	77,5	90,0	90,0
TGS2620	75,0	80,0	90,0

**Tabla 3-16: % de acierto problema de días peculiares**

Los resultados obtenidos por *Random Forest* basado en SMTS no son positivos, teniendo en cuenta que el 25% de fallo corresponde a los cinco datos de días peculiares, acertando muy

de vez en cuando uno de ellos, de donde se obtiene el 77,5% de acierto de dos de los sensores. Estos resultados son iguales, aunque traslademos uno o dos de los casos peculiares al conjunto de entrenamiento, o, aunque reduzcamos dicho conjunto de tamaño.

Sin embargo, las redes convolucionales obtienen resultados mucho mejores, alcanzando el 100% de acierto utilizando o todos los atributos o únicamente CO<sub>2</sub>. Hay que especificar que este resultado no se obtiene en todas las épocas, pero aun así la mayoría superan el 75%. Esto se debe a que las redes convolucionales son capaces de generalizar mejor ya que su diseño busca evitar el sobreajuste.

### 3.7 Problema de horas de la limpieza

En este problema queremos clasificar franjas de una hora en base a si se realizó o no trabajo de limpieza, teniendo por tanto dos clases, disponiendo de 32 series para cada una. A diferencia del resto de problemas, en este concretamente podemos esperar una mayor eficacia en base a los sensores TGS frente al sensor de CO<sub>2</sub> ya que diversos productos químicos son utilizados durante la limpieza de aulas.

	SMTS	FCN	RNN
Todos	62,3	86,6	83,3
CO <sub>2</sub>	64	79,72	79,86
Humedad	61	73	66,52
TGS2600	68	93,32	96,6
TGS2602	74	93,32	86,4
TGS2603	71	93,32	83,26
TGS2610	68	86,52	93,32
TGS2611	69,3	93,32	93,32
TGS2620	75	89,96	89,96

**Tabla 3-17: % de acierto problema de horas de la limpieza**

Como esperábamos, los distintos sensores TGS obtienen mejores resultados en comparación a la clasificación por CO<sub>2</sub>, sin destacar ninguno en especial de ellos por encima del resto. También es destacable que al igual que en el problema anterior, *Random Forest* basado en SMTS obtiene resultados notablemente peores que las redes neuronales, probablemente debido tanto a una cantidad de datos mejorable como a patrones no tan claros en las horas de la limpieza, por lo que la dificultad a la hora de generalizar que tiene SMTS se vuelve relevante.

## 4 Conclusiones y discusión

---

### 4.1 Conclusiones

En este trabajo hemos utilizado un conjunto de datos provenientes de una nariz artificial situada en una clase de un colegio de primaria para poder detectar y clasificar rutinas humanas mediante aprendizaje automático a partir de la monitorización no invasiva de la nariz. Las series de este conjunto son multivariadas y temporales, correspondientes a cada uno de los sensores de la nariz, lo que convierte esta detección en un problema no trivial.

La primera tarea que hemos realizado en este trabajo ha consistido en asignar etiquetas a distintas series temporales correspondientes a horas o días en función de las distintas características de la franja temporal a la que corresponden, como puede ser el día de la semana, el horario de inicio de la actividad y el tipo de actividad. Con cada etiquetado, creamos un problema de clasificación distinto con el que realizar pruebas a partir de los datos disponibles.

Hay que mencionar que los resultados obtenidos durante nuestras pruebas son representativos para las configuraciones de los clasificadores que hemos usado, pudiendo variar si se modificase alguno de los parámetros seleccionados a priori.

El primer problema, el más simple, consistía en diferenciar entre días lectivos y no lectivos. Los resultados son muy positivos, llegando a clasificar todos los datos de los que disponemos correctamente. De manera más concreta, el siguiente problema consistió en diferenciar periodos lectivos de no lectivos focalizando el estudio en periodos no lectivos colindantes a lectivos. En este problema obtenemos también buenos resultados, pero encontramos horas que no se clasifican bien, debido principalmente a horas lectivas con registro de los sensores peculiarmente bajos.

A continuación, hemos abordado el primer problema con más de dos clases. Este problema consiste en detectar el día de la semana lectivo al que corresponde una serie. Aquí ya nos encontramos con menor precisión, rondando el 50% de acierto, pero aun así los resultados no son malos, teniendo en cuenta que nos encontramos frente a 5 clases distintas y con series temporales con deriva, dependencia de la historia previa y alta variabilidad. De entre todos los días, miércoles y viernes son los más complicados de clasificar. No podemos concretar la causa de esto ya que la distribución de asignaturas es totalmente distinta para cada día, sin poder observar nada que pudiese dificultar la clasificación de estos dos días de forma particular más allá de la diversidad de actividades y su orden diario específico.

A continuación, intentamos identificar la asignatura a la que corresponde cada serie. Este problema de 8 clases distintas es aún más complejo, y los resultados dejan que desear, con una precisión del 30%, siendo casi incapaz de identificar la asignatura de valores a partir del conjunto de entrenamiento y de los parámetros de los clasificadores utilizados. Aun así, mediante un análisis individual hemos podido observar que los clasificadores son capaces de distinguir correctamente varios segmentos de las series temporales correspondientes a asignaturas específicas, lo que apunta a la posibilidad de este tipo de clasificación con modelos más optimizados.

Otro de los problemas abordados ha consistido en identificar la franja horaria de una serie entre las 6 de las que consta cada día lectivo. Se supera el 50% de acierto, principalmente gracias a la primera franja, aunque para el resto también se obtienen resultados que denotan diferencias apreciables.

Para acabar con el estudio de este conjunto de datos, intentamos identificar los días peculiares de entre todos los lectivos, siendo estos días los que hemos confirmado que tuvieron eventos extraordinarios como excursiones o charlas, que afectan mucho la actividad en el aula. Debido a la baja cantidad de este tipo de días durante el curso, incluimos algunos días no lectivos en el conjunto como días peculiares para el entrenamiento. Aunque llegamos a obtener un 100% de precisión, cabe recalcar que la cantidad de datos es baja y sería deseable validar este estudio con un mayor número de patrones para dar credibilidad completa a los resultados.

Por otro lado, en base a un conjunto de datos de otras narices situadas en aulas de la Escuela Politécnica Superior de la UAM, intentamos diferenciar entre horas en las que había trabajo de limpieza y horas en las que no existía ese tipo de actividad. Los resultados muestran que los gases detectados durante esta actividad por los sensores TGS son mucho más determinantes que en el resto de los problemas, y son capaces de clasificar la mayoría de las series con un acierto medio que ronda el 90% mediante redes neuronales profundas.

Con respecto a los clasificadores, hemos podido observar que el *Random Forest* basado en SMTS ha obtenido resultados ligeramente mejores que aquellos obtenidos por los algoritmos derivados de red convolucional en los problemas donde disponemos de suficientes datos. Sin embargo, en los dos problemas en los que tenemos un número limitado de datos, por ejemplo, en el problema de días peculiares y el problema de días de la limpieza, sus resultados son notablemente peores. Con esto podemos concluir que el *Random Forest* es capaz de ser más preciso con una suficiente cantidad de datos, pero no tiene la capacidad de generalizar de la que constan los algoritmos derivados de redes convolucionales a partir de un número reducido de ejemplos.

De entre todas las medidas registradas por los sensores de las narices artificiales, el nivel de CO<sub>2</sub> ha destacado en la mayoría de los problemas como el más informativo a excepción del problema de días de la limpieza. De los sensores TGS, el sensor TGS2603 (series de aminas, olores sulfurosos), ha destacado ligeramente sobre el resto durante las pruebas.

En conclusión, se pueden observar, aunque con cierta dificultad en algunas de las actividades, patrones en la estructura temporal de los niveles de distintos gases que ayudan a reconocer y diferenciar rutinas humanas de forma no invasiva. Aun así, no es una tarea fácil, ya que no son problemas triviales, y en varios de los ejemplos mostrados se puede observar que la habilidad para reconocer las mencionadas rutinas puede ser limitada con las medidas y la parametrización de los clasificadores que hemos utilizado.

## **4.2 Trabajo futuro**

En este trabajo puede ser extendido de diversas formas. Para empezar, se puede explorar más en profundidad el problema de los días peculiares, ya que los datos que hemos utilizado para ello son escasos. De forma similar, el mencionado problema de las horas de examen también podría estudiarse con una cantidad superior de datos.

Los problemas de los que se han obtenido resultados mejorables también podrían ser explorados más a fondo buscando otras medidas útiles derivadas de la presencia de eventos secuenciales entre señales de sensores, o simplemente estudiando en profundidad cada uno individualmente.

Sobre los algoritmos de aprendizaje empleados en este estudio, se pueden explorar valores de parámetros distintos a los utilizados, buscando para cada problema la mejor combinación posible. En los algoritmos derivados de redes convolucionales se puede intentar modificar la arquitectura de cada uno, cambiando el contenido de los bloques, alterando el número de filtros de las convoluciones, o modificando el tamaño y cantidad de *kernels*. En base a nuestras pruebas, un número mayor de épocas al finalmente utilizado no parece reportar mejores resultados, pero quizás sea posible optimizar los resultados mediante la modificación de la arquitectura, al igual que el tamaño de lote. En el caso del *Random Forest* basado en SMTS, se puede explorar un mayor abanico de valores para  $R$  (Símbolos del alfabeto) o  $J_{ins}$  (Número de árboles de decisión del *Random Forest* de clasificación). Por otro lado, también se podría explorar el efecto de la codificación temporal mediante el estudio del valor óptimo para la diferencia temporal utilizada. Es decir, en vez de utilizar la diferencia entre valores consecutivos para la codificación, se podría utilizar un valor  $\tau$  a explorar en la siguiente representación de la codificación:

$$[t, x_1^n(t), x_1^n(t) - x_1^n(t - \tau), x_2^n(t), \dots, x_m^n(t), x_m^n(t) - x_m^n(t - \tau)]$$

Finalmente, teniendo en cuenta que el objetivo de este trabajo es la monitorización de rutinas humanas, también se pueden explorar de manera similar muchos otros problemas en distintos ámbitos, no solo en el escolar como hemos profundizado aquí.



# Referencias

---

- Abanda, A., Mori, U., and Lozano, J. A. (2019). A review on distance based time series classification. *Data Min. Knowl. Discov.* 33, 378–412. doi:10.1007/s10618-018-0596-4.
- Andersen, T. G., Davis, R. A., Kreiß, J.-P., and Mikosch, T. V (2009). *Handbook of financial time series*. Springer Science & Business Media.
- Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Discov.* 31, 606–660. doi:10.1007/s10618-016-0483-9.
- Baydogan, M. G., and Runger, G. (2014). Learning a symbolic representation for multivariate time series classification. *Data Min. Knowl. Discov.* 29, 400–422. doi:10.1007/s10618-014-0349-y.
- Choi, K., Fazekas, G., and Sandler, M. (2016). Automatic tagging using deep convolutional neural networks. *Proc. 17th Int. Soc. Music Inf. Retr. Conf. ISMIR 2016*, 805–811.
- Ding, J., McAvoy, T. J., Cavicchi, R. E., and Semancik, S. (2001). Surface state trapping models for SnO<sub>2</sub>-based microhotplate sensors. *Sensors Actuators, B Chem.* 77. doi:10.1016/S0925-4005(01)00765-1.
- Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Min. Knowl. Discov.* 33, 917–963.
- Herrero-Carrón, F., Yáñez, D. J., Rodríguez, F. de B., and Varona, P. (2015). An active, inverse temperature modulation strategy for single sensor odorant classification. *Sensors Actuators B Chem.* 206, 555–563. doi:https://doi.org/10.1016/j.snb.2014.09.085.
- Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P. A. (2019). Deep learning for time series classification: a review. *Data Min. Knowl. Discov.* 33, 917–963. doi:10.1007/s10618-019-00619-1.
- Martinelli, E., Magna, G., De Vito, S., Di Fuccio, R., Di Francia, G., Vergara, A., et al. (2013). An adaptive classification model based on the Artificial Immune System for chemical sensor drift mitigation. *Sensors Actuators, B Chem.* 177, 1017–1026. doi:10.1016/j.snb.2012.11.107.
- Montgomery, D. C., Jennings, C. L., and Kulahci, M. (2015). *Introduction to time series analysis and forecasting*. John Wiley & Sons.
- Rozado, D., Rodriguez, F. B., and Varona, P. (2010). Optimizing Hierarchical Temporal Memory for Multivariable Time Series. *Lect. Notes Comput. Sci.* 6353, 506–518. doi:10.1007/978-3-642-15822-3\_62.
- Rozado, D., Rodriguez, F. B., and Varona, P. (2012). Low cost remote gaze gesture recognition in real time. *Appl. Soft Comput.* 12, 2072–2084.
- Small, M. (2005). *Applied nonlinear time series analysis: applications in physics, physiology and finance*. World Scientific.
- Walt, D. R., Stitzel, S. E., and Aernecke, M. J. (2012). Artificial noses. *Am. Sci.* 100, 38–45. doi:10.1146/annurev-bioeng-071910-124633.
- Wan, R., Mei, S., Wang, J., Liu, M., and Yang, F. (2019). Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting. *Electron.* 8. doi:10.3390/electronics8080876.
- Wang, Z., Yan, W., and Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. in *Proceedings of the International Joint*

- Conference on Neural Networks* (Institute of Electrical and Electronics Engineers Inc.), 1578–1585. doi:10.1109/IJCNN.2017.7966039.
- Ye, L., and Keogh, E. (2011). Time series shapelets: A novel technique that allows accurate, interpretable and fast classification. *Data Min. Knowl. Discov.* 22, 149–182. doi:10.1007/s10618-010-0179-5.
- Yuan, Z., Han, E., Meng, F., and Zuo, K. (2020). Detection and Identification of Volatile Organic Compounds Based on Temperature-Modulated ZnO Sensors. *IEEE Trans. Instrum. Meas.* 69, 4533–4544. doi:10.1109/TIM.2019.2948413.
- Zeger, S. L., Irizarry, R., and Peng, R. D. (2006). On time series analysis of public health and biomedical data. *Annu. Rev. Public Heal.* 27, 57–79.
- Zhao, B., Lu, H., Chen, S., Liu, J., and Wu, D. (2017). Convolutional neural networks for time series classification. *J. Syst. Eng. Electron.* 28, 162–169.
- Zohora, S. E., Khan, A. M., and Hundewale, N. (2013). “Chemical sensors employed in electronic noses: A review,” in *Advances in Computing and Information Technology* (Springer), 177–184.



## Glosario

---

MTS	Multivariate Time Series
SMTS	Symbolic representation for Multivariate Time Series
FCN	Fully Convolutional Network
RNN	Residual Neural Network
TGS	Taguchi Gas Sensor
TGS2600	Sensor de contaminantes en el aire. Hidrógeno y CO.
TGS2602	Sensor de contaminantes en el aire. Amoniacio y H <sub>2</sub> S.
TGS2603	Sensor de aminas
TGS2610	Sensor de gas LP (licuado de petróleo).
TGS2611	Sensor de metano (CH <sub>4</sub> ).
TGS2620	Sensor de vapors orgánicos.

## **Anexo A: Calendario escolar**

---

Calendario escolar provisto por el centro para el etiquetado de las series temporales.

<b>Hora</b>	<b>Lunes</b>	<b>Martes</b>	<b>Miércoles</b>	<b>Jueves</b>	<b>Viernes</b>
9:00 - 9:45	Lengua	E.F.	Mates	Valores	Mates
9:45 - 10:30	C.Naturales	Inglés	C.Naturales	Música	Lengua
10:30 - 11:15	Mates	Mates	C.Sociales	E.F.	Inglés
11:15 - 11:45	Recreo				
11:45 - 12:30	Mates	Valores	Inglés	Lengua	E.F.
12:30 - 13:15	Inglés	Lengua	Mates	Mates	C.Naturales
13:15 - 15:15	Hora de comer				
15:15 - 16:00	Sociales	Música	Lengua	Inglés	Lengua

E.F: Educación Física

Mates: Matemáticas

C.Naturales: Ciencias naturales

C.Sociales: Ciencias sociales